Text to Audio Converter & Summarizer: A MERN Stack Project with AI Integration

By – Pawan Raj , Neha Kumari , Guided by – Rakesh Jaiswal

Department :- Computer Science and Business System ,

Oriental Institute Of Science And Technology , Bhopal

Abstract:

The increasing need for accessible and efficient content consumption has drastically propelled advancements in AI-powered applications, especially in the area of natural language processing and speech synthesis. This research paper explores the creation of a cutting-edge Text to Audio Converter & Summarizer developed with the MERN (MongoDB, Express.js, React.js, Node.js) stack and then augmented with advanced Artificial Intelligence (AI) integration to enhance the user experience. The main aim of this project is to convert huge amounts of text into concise and natural-sounding audio while also creating abridged and meaningful summaries. This is done through utilizing the latest AI models like BART and T5 for summarization, coupled with high-capacity Text-to-Speech (TTS) engines like Google Text-to-Speech (gTTS) and Coqui TTS. The paper elaborately describes the architecture of the system, the technologies used, the step-by-step implementation process, issues encountered during development, and recommendations for further enhancements directed at increasing accessibility and usability.

---

1. Introduction

In the fast-paced digital age of today, there is increasing demand for effective and convenient ways of processing huge quantities of textual data in a matter of minutes. Conventional reading may be time-consuming and inaccessible to the visually impaired or auditory learners. The challenges are overcome by text-to-audio transformation and summarization technologies that offer users audio copies of the text and shorter summaries of the key points. This work describes the creation of a complete web application that combines these features based on the MERN stack—MongoDB, Express.js, React.js, and Node.js—with state-of-the-art Artificial Intelligence (AI) models. Through the use of advanced Natural Language Processing (NLP) methods and cutting-edge Text-to-Speech (TTS) engines, this app will revolutionize the way users interact with text content, making it more interactive and simple to use. The system concentrates on providing swift, precise, and natural-speech speech along with informative summaries for enhanced overall user experience.

MERN Stack:

- MongoDB: A NoSQL database for storing text inputs, summaries generated, and audio files.
- Express.js: A backend framework for developing the API endpoints for handling user requests.
- React.js: A frontend framework utilized for developing a dynamic and user-friendly interface.

• Node.js: JavaScript runtime environment used to run server-side code and manage interaction between frontend and backend.

AI & NLP Integration:

• BART and T5 Models: Used for text summarization, these AI models take in input text, process it, and convert it into intelligent summaries.

• Google Text-to-Speech (gTTS) & Coqui TTS: These state-of-the-art Text-to-Speech (TTS) engines translate text to natural-sounding speech.

• Google Translation API: Translates text into several languages to increase accessibility for the global audience.

3. System Architecture

The system is based on a client-server architecture, in which:

• Frontend (React.js): The user interacts with the interface to type text, choose summarization or audio conversion mode, and get the corresponding output (audio or summary).

• Backend (Node.js & Express.js): Processes user requests, communicates with AI models for summarization and speech synthesis, and performs data operations.

• Database (MongoDB): Holds user information, text inputs, created summaries, and audio files.

• AI Integration: Input text is processed through AI models by the summarization and TTS modules for precise output.

4. Implementation Details

• Text Preprocessing:

The initial step is to prepare the input text by discarding unnecessary portions, such as stop words, punctuation, and other noise with NLP libraries. This process aids in enhancing the summarization process accuracy.

• Summarization Model:

BART or T5 models run the preprocessed text and summarize the key points to provide a brief summary, making the content more readable.

• Text-to-Speech Conversion:

The processed or summarized text is then synthesized into speech through the use of powerful TTS engines such as Google TTS or Coqui TTS. These TTS models provide natural-sounding voice output, simulating human voice tones and pace.

• Translation:

The Google Translation API is incorporated to enable the users to translate the input text into other languages prior to processing for summarization and speech synthesis.

• Features for Frontend:

User interface, created using React.js, enables the users to enter text, choose summarization settings, translate text, play the audio played after conversion, and download the created files. The interface is intuitive and responsive for various devices.

5. Challenges & Solutions

• Latency Issues:

AI model response times can add latency, particularly when dealing with extensive text inputs. Response time can be optimized by caching model inference and enhancing server performance using multi-threading or cloud-based options.

• Speech Quality:

Natural-sounding speech output can be tricky to maintain. To counter this, advanced TTS models such as Google TTS and Coqui TTS with their customizable voices and enhanced prosody are utilized.

• Data Management:

Effective storage and retrieval of text, summaries, and audio files are essential. MongoDB database is designed to support large datasets, and file management is processed through cloud storage solutions for scalability and redundancy.

6. Future Improvements

•Multi-Language Support:

Diversification of the application to support additional number of multiple languages will allow users from various regions to use the system.

•Voice Modulation and Customizable Speech Tones:

The feature to select various voices, accents, and tones will make the user experience better and according to individual tastes.

•Android application version development:

As it is a web application therefore developing an android application version for our project will make it even more scalable and easy to use even in offline mode which results in the greater number of users and creating a user-friendly environment.

7. Conclusion

The effective implementation of Artificial Intelligence in a MERN stack-based Text to Audio Converter & Summarizer provides a promising answer towards increasing content accessibility and greatly boosting user interaction. Through the synergy of cutting-edge Natural Language Processing (NLP) methods for text summarization with innovative speech synthesis technologies like Google TTS and Coqui TTS, the project offers a fast and intuitive platform for converting written text into natural and understandable speech. The fact that the Google Translation API is incorporated means that the application becomes usable across various languages, thus opening its use to a global multilingual user base. The project has a solid foundation on which future development is based, such as extending multi-language support, adding customizable voice output, and creating mobile apps to enhance usability and scalability. These enhancements will ultimately help make it easier for content to be consumed and accessed by users globally.

References

1. Vaswani, A., et al. (2017). Attention Is All You Need. NeurIPS.
2. Raffel, C., et al. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (T5). JMLR.
3. Shen, J., et al. (2018). Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. IEEE.
4. 6. Zenkel, T., et al. (2020). End-to-End Speech Translation with Neural Attention. ICASSP.
5. 7. He, K., et al. (2016). Deep Residual Learning for Image Recognition. CVPR.
6. 8. Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. EMNLP. https://aclanthology.org/D15-1166/
7. 9. Mozilla TTS Documentation. (2023).
8. 10. Google Cloud Text-to-Speech API. (2024).
9. 11. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. NeurIPS.
10. 12. Tan, Y., & Celikyilmaz, A. (2020). Summarization Transformers: Text Summarization with Pretrained Encoders. arXiv.
11. 16. Wang, Y., et al. (2017). Tacotron: Towards End-to-End Speech Synthesis. Interspeech.
12. 17. Oord, A. v. d., et al. (2016). WaveNet: A Generative Model for Raw Audio. SSW.
13. 18. Li, J., et al. (2020). Transformer-Based Text Summarization with Reinforcement Learning. ACL.
14. 19. Kannan, A., et al. (2019). Smart Compose: A Neural Network Based Writing Assistant. Interspeech.
15. 20. Papineni, K., et al. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. ACL.
16. 21. Li, H., et al. (2023). A Survey on Text-to-Speech Synthesis: Models, Datasets and Challenges. IEEE Access.